Remarks

In the following *Remarks*, Applicants will first traverse Examiner's rejections of claims 5, 9-14, 19, 20, 22-28 as anticipated by Collberg, et al., *A Taxonomy of obfuscating transformations* (Henceforth Collberg, *Taxonomy*) and will then demonstrate the patentability of Applicants' claims as amended over Collberg, et al., *Software watermarking: Models and dynamic embeddings*, published in January of 1999 (henceforth Collberg, *Software watermarking).*

Traversal of Examiner's rejections

The difficulties with Examiner's rejection of claims 1-4, 9-14, 19, 20, and 22-28 as anticipated by Collberg, *Taxonomy* can be seen by comparing the title of Applicants' patent application with the full title of the reference. The title of the application is *Authenticating executable code and executions thereof*; the full title of the reference is *A Taxonomy of Obfuscating Transformations*. About all the references have in common with each other is the following: the transformations used to add a static watermark to executable code and those used to obfuscate the executable code must both have the property that the transformed code must be equivalent to the untransformed code for the purpose for which the untransformed code was originally intended. The purposes of the transformations are, however, completely different in the two cases: in the case of Applicants' inventions, the purpose of the transformation is to watermark the code with a watermark value that can be used to determine the code's authenticity. In the case of Collberg, *Taxonomy*, the purpose of the transformation is to make it difficult to reverse engineer the code. As would be expected from this circumstance, searches in Collberg, *Taxonomy* on the terms "watermark" and "key", and "authentic", all of which are important terms in Applicants' disclosure, found nothing.

Continuing with a detailed analysis of Examiner's rejections, claim 1 is rejected on the basis of Figs. 1, 5, and 6 and page 3, col. 2 through page 7, column 1, top. Beginning with the claim step "receiving the sequence of executable instructions and a key", there is, as pointed out above, no mention of a key on page 3, col. 2, in figures 1 or 5, or as pointed out above, anywhere else in the Collberg, *Taxonomy* reference. Since that is the case, there can also be no disclosure of anything like the limitation in the second step of "modifying the sequence of executable instruction in a manner determined by the key ". Since that is so, there is no basis for the rejection of claim 1 as anticipated by Collberg, *Taxonomy.*

Claims 2-17 are of course all patentable over Collberg, *Taxonomy* because they are dependent from claim 1, which is patentable over the reference, but they are also all patentable over the reference in their own rights. Claim 2 sets forth that a watermark value is received along with the key and the sequence of executable instructions; Collberg, *Taxonomy* makes no mention of a watermark value in the cited locations or anywhere else in the reference. Claims 3-5 set forth how the key is used to determine the locations at which the sequence is to be modified to represent the watermark value and details of the watermark; these claims cannot be anticipated by a reference that says nothing about watermarks. In claims 6, 7, and 15-17, the watermark value is provided to an authenticating entity. Collberg, *Taxonomy*, discloses nothing whatever about authentication or an authenticating entity.

Claims 8-16 concern making a watermark by modifying the sequence of executable instructions "such that when the modified sequence of executable instructions in executed, execution state is produced which has a property that depends on the key" (claim 8). Collberg, *Taxonomy* is about changing the way code *looks* so that it is hard to reverse engineer, not about changing the way code *behaves* so that "execution state is produced which has a property that depends on the key", and there is simply no disclosure whatever about execution state at all, let alone about execution state "which has a property that depends on a key".

The remaining independent claims are claims 18 and 21. Both of these claims are addressed to "methods of authentication", concerning which there is no disclosure in Collberg, *Taxonomy.* Claim 18 further involves a watermarked sequence, a key, and a watermark value, none of which is disclosed in Collberg, *Taxonomy.* Claim 21 further includes "a watermarked sequence", a key, execution state, and a description of the execution state, none of which is disclosed in Collberg, *Taxonomy*.

As may be seen from the foregoing, Collberg, *Taxonomy* fails to disclose all of the limitations of any of claims 1-4, 6-8, 15-18, or 21, and for that reason, the reference anticipates none of these claims and the Examiner's rejection of the claims under 35 U.S.C. 102 is without basis.

Patentability of the claims over the references cited in the IDS mailed 6/24/04

Collberg references having to do with the watermarking of executable code

While searching the Web to find an electronic copy of the Collberg, *Taxonomy* reference, Applicants' attorney found that Collberg had also addressed the watermarking of executable code. The following references appear in a bibliography provided by Dr. Collberg at www.cs.arizona.edu:

- 10 June 1998: New Zealand patent application 330675, Collberg, et al., *Software watermarking techniques,* from which the PCT application PCT/NZ99/00081 claims priority.
- 26 August 1998: Collberg, et al., *On the limits of Software Watermarking*, a paper which appears to have substantially the same disclosure as the patent application.
- January 1999: Collberg, et al., *Software watermarking: Models and Dynamic Embeddings*, which "supersedes" *On the limits of software watermarking.*
- 10 June 1999: PCT/NZ99/00081, Collberg, et al., *Software watermarking Techniques,* claiming priority from NZ 330675 and designating the US. It appears that no US published patent application or patent corresponding to this PCT application has yet issued.
- 16 December 1999: PCT/NZ99/00081 published as WO 99/64973

The priority date for Applicants' patent application is 12 May, 1999, and Applicants have amended their claims on the presumption that Collberg, *Software Watermarking* is prior art to Applicants' inventions. Applicants' date of conception may, however, be earlier than 26 August 1998 or 10 June 1998. Should that turn out to be the case, Applicants reserve the right to file proof of Applicants' date of conception and to amend their claims as permitted by the earlier date of conception.

Disclosures of the Collberg references

The most comprehensive of the Collberg references is the Collberg, *Software watermarking* reference. This reference will be treated in the following as representative for them all. In *Software watermarking,* the authors are concerned with using watermarks in software to identify the owner of the software:

> Watermarking embeds a secret message into a cover message. In media watermarking the secret is usually a copyright notice and the cover a digital image. Watermarking an object discourages intellectual property theft, or when such theft has occurred, allows us to prove ownership. (Collberg, *Software watermarking*, Abstract)

There is simply no notion in Collberg, *Software Watermarking* that a watermark might be used to *authenticate* the software. The character string "authentic" does not appear in the reference. The closest that Collberg, *Software watermarking* comes to the topic of authentication is the discussions of tamperproofing watermarks in sections 2.3 and 5.5 and 5.5.1. Tamperproofing a watermark is taking measures to make sure that the watermark cannot be removed from the code and this, of course, is not the same as tamperproofing the code that carries the watermark. Section 5.5.1 does include one example which shows Java's class reflection mechanism may be used to detect tampering with a watermark based on a graph's node type and the program may be made to terminate on detection.

What Collberg, *Software watermarking* is chiefly concerned with is the difficulty of watermarking something which is as malleable as executable code. For example, static watermarks in executable code can be removed simply by obfuscating the executable code until the watermark is no longer detectable (see Section 2.3). For that reason, Collberg, *Software watermarking* prefers dynamic watermarking, but even there, obfuscation can render most dynamic watermarks undetectable (see Section 5.) In

Section 5, Collberg, *Software watermarking* discloses a new watermarking technique called "dynamic graph watermarking" which has more resistance to obfuscation attacks. The conclusion which must be drawn from Collberg, *Software watermarking* is that only the most complex dynamic watermarking techniques are of any use in protecting executable code.

Watermarking and authentication

While Collberg's pessimism about watermarking executable code may be justified when the watermark is used to show ownership, Applicants have demonstrated that even simple static watermarking is an effective way to authenticate executable code. The reason that this is so is that in authentication, loss or corruption of the watermark is *proof that the code has changed since it was watermarked.* Thus, the very property of a watermark in code that renders the watermark almost useless for showing ownership makes the watermark extremely useful for detecting faulty transmission of the code or tampering with the code and therefore for authenticating the code.

Applicants' claims

Applicants' original claims 6, 7, 15-17 and 18-28 are all addressed to the use of watermarks in code to authenticate the code. Because Collberg, *Software watermarking,* discloses nothing at all about using watermarks to authenticate code, it cannot anticipate these original claims. Applicants have further amended claim 1 to clearly point out that the method adds the watermark to a sequence of executable instructions "to render the sequence authenticatable" and that the watermark represents "a watermark value which may be employed to authenticate the sequence".

1. (presently amended) A method of adding a watermark to a sequence of executable instructions to render the sequence authenticatable, the method comprising the steps of:

receiving the sequence of executable instructions and a key; and

using the key to modify~~ing~~ the sequence of executable instructions ~~in a manner determined by the key~~so that the watermark may be obtained from the modified sequence, the sequence being modified such that the usefulness of the modified sequence for the sequence's intended purpose is not affected by the modifications made thereto and the watermark representing a watermark value which may be employed to authenticate the sequence.~~, the sequence being modified such that the usefulness of the sequence for the sequence's intended purpose is not affected thereby.~~

As amended, claim 1 is patentable over Collberg, *Software watermarking* for the same reasons that claims 6,7, 15-17, and 18-28 are patentable over the reference. The amendments to the claims dependent from claim 1 clarify the claims and bring them into conformity with claim 1 as amended. Examiner will immediately see that the amendment is fully supported by the Specification as filed.

Conclusion

Applicants are including a petition for a one-month extension of time with this response, have accepted Examiner's renumbering of their claims, have traversed Examiner's rejection of their claims as anticipated by Collberg, *Taxonomy,* have disclosed the newly-found Collberg references concerning watermarking of code to Examiner, have amended claim 1 to distinguish it from Collberg, *Software watermarking,* and have demonstrated that claims 6,7, 15-17, and 18-28 as originally filed were patentable over Collberg, *Software watermarking*. Applicants have thus been completely responsive to Examiner's Office action of 3/11/2004 as required by 37 C.F.R. 1.111(b) and respectfully request that Examiner continue with his examination of the application as provided by 37 C.F.R. 1.111(a).

A check for $55.00 for a one-month extension of time accompanies this response; no other fees are believed to be required for this response. Should any under- or overpayments have been made, please charge any additional fees required for the amendment or refund any overpayments to deposit account number 501315.

Respectfully submitted,
Gordon E Nelson
Attorney of record,
Gordon E. Nelson
57 Central St., P.O. Box 782
Rowley, MA, 01969,
Registration number 30,093
Voice: (978) 948-7632
Fax: (866)-723-0359
6/29/04
Date

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

on 6/29/04
(Date)
Gordon E. Nelson, #30,093
Gordon E Nelson
(Signature)